

Deploying Sphinx documentation to GitHub Pages

📌 Objectives

- Create a basic workflow which you can take home and adapt for your project.

GitHub Pages

- Serve websites from a GitHub repository.
- It is no problem to serve using your own URL `https://myproject.org` instead of `https://myuser.github.io/myproject`.

GitHub Actions

- Automatically runs code when your repository changes.
- We will let it run `sphinx-build` and make the result available to GitHub Pages.

Our goal: putting it all together

- Host source code with documentation sources on a public Git repository.
- Each time we `git push` to the repository, a GitHub action triggers to rebuild the documentation.
- The documentation is pushed to a separate branch called 'gh-pages'.

Exercise - Deploy Sphinx documentation to GitHub Pages

🔧 GH-Pages-1: Deploy Sphinx documentation to GitHub Pages

In this exercise we will create an example repository on GitHub and deploy it to GitHub Pages.

Step 1: Go to the [documentation-example](#) project template on GitHub and create a copy to your namespace.

- Give it a name, for instance “documentation-example”.
- You don’t need to “Include all branches”
- Click on “Create a repository”.

Step 2: Browse the new repository.

- It will look very familiar to the previous episode.
- However, we have moved the documentation part under `doc/` (many projects do it this way). But it is still a Sphinx documentation project.
- The source code for your project could then go under `src/`.

Step 3: Add the GitHub Action to your new Git repository.

- Add a new file at `.github/workflows/documentation.yml` (either through terminal or web interface), containing:

```
name: documentation

on: [push, pull_request, workflow_dispatch]

permissions:
  contents: write

jobs:
  docs:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-python@v5
      - name: Install dependencies
        run: |
          pip install sphinx sphinx_rtd_theme myst_parser
      - name: Sphinx build
        run: |
          sphinx-build doc _build
      - name: Deploy to GitHub Pages
        uses: peaceiris/actions-gh-pages@v3
        if: ${ github.event_name == 'push' && github.ref == 'refs/heads/main' }}
        with:
          publish_branch: gh-pages
          github_token: ${ secrets.GITHUB_TOKEN }}
          publish_dir: _build/
          force_orphan: true
```

- You don’t need to understand all of the above, but you might spot familiar commands in the `run:` sections.

- After the file has been committed (and pushed), check the action at `https://github.com/USER/documentation-example/actions` (replace `USER` with your GitHub username).

Step 4: Enable GitHub Pages

- On GitHub go to “Settings” -> “Pages”.
- In the “Source” section, choose “Deploy from a branch” in the dropdown menu.
- In the “Branch” section choose “gh-pages” and “/root” in the dropdown menus and click save.
- You should now be able to verify the pages deployment in the Actions list).

Step 5: Verify the result

- Your site should now be live on `https://USER.github.io/documentation-example/` (replace `USER`).

Step 6: Verify refreshing the documentation

- Commit some changes to your documentation
- Verify that the documentation website refreshes after your changes (can take few seconds or a minute)

Alternatives to GitHub Pages

- [GitLab Pages](#) and [GitLab CI](#) can create a very similar workflow.
- [Read the Docs](#) is the most common alternative to hosting in GitHub Pages.
- Sphinx builds HTML files (this is what static site generators do), and you can host them anywhere, for example your university’s web space or own web server.

Migrating your own documentation to Sphinx

- First convert your documentation to Markdown using [Pandoc](#).
 - Create a file `index.rst` which lists all other Markdown files and provides the table of contents.
 - Add a `conf.py` file. You can generate a starting point for `conf.py` and `index.rst` with `sphinx-quickstart`, or you can take the examples in this lesson as inspiration.
 - Test building the documentation locally with `sphinx-build`.
 - Once this works, follow the above steps to build and deploy to GitHub Pages or some other web space.
-

! Keypoints

- Sphinx makes simple HTML (and more) files, so it is easy to find a place to host them.
- Github Pages + Github Actions provides a convenient way to make sites and host them on the web.